Apache Oozie User Guide

Table of Contents

What is Apache Oozie?

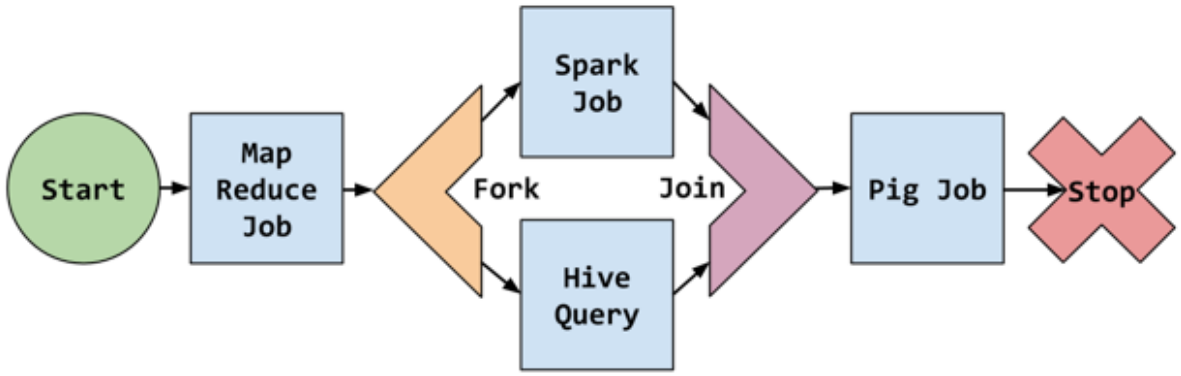
Oozie is a workflow scheduler system to manage Apache Hadoop jobs. Oozie Workflow jobs are Directed Acyclic Graphs (DAGs) of actions.

Oozie is integrated with the rest of the Hadoop stack supporting several types of Hadoop jobs out of the box (such as Java map-reduce, Streaming map-reduce, Pig, Hive, Sqoop and Distcp) as well as system specific jobs (such as Java programs and shell scripts). Oozie is a scalable, reliable and extensible system. Oozie workflows contain control flow nodes and action nodes.

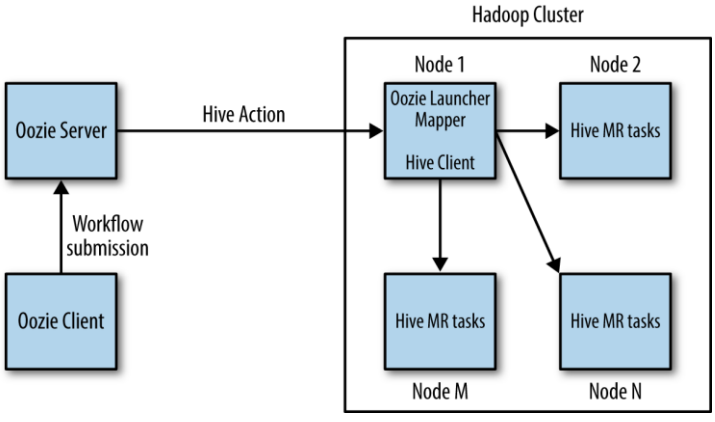
Control flow nodes define the beginning and the end of a workflow (start, end and fail nodes) and provide a mechanism to control the workflow execution path (decision, fork and join nodes).

Action nodes are the mechanism by which a workflow triggers the execution of a computation/processing task. Oozie provides support for different types of actions: Hadoop map-reduce, Hadoop file system, Pig, SSH, HTTP, eMail and Oozie sub-workflow. Oozie can be extended to support additional type of actions.

Oozie workflows can be parameterized (using variables like ${inputDir} within the workflow definition). When submitting a workflow job values for the parameters must be provided. If properly parameterized (i.e. using different output directories) several identical workflow jobs can concurrently.



Apache Oozie Architecture



What is Oozie Coordinator?

Oozie Coordinator jobs are recurrent Oozie Workflow jobs triggered by time (frequency) and data availability.

What is Oozie Bundle?

Bundle is a higher-level oozie abstraction that will batch a set of coordinator applications. The user will be able to start/stop/suspend/resume/rerun in the bundle level resulting a better and easy operational control.

More specifically, the oozie Bundle system allows the user to define and execute a bunch of coordinator applications often called a data pipeline. There is no explicit dependency among the coordinator applications in a bundle. However, a user could use the data dependency of coordinator applications to create an implicit data application pipeline.

Oozie Workflow Simple Example

Table create script for hive table:

```
CREATE TABLE IF NOT EXISTS customers (id int, name String, age int, address String, salary double) COMMENT 'customer details' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE;
```

Table insert script for hive table:

```
LOAD DATA INPATH '${INPUT_PATH}' INTO TABLE customers;
```

job.properties

```
nameNode=hdfs://instance-1.us-east1-b.c.nifty-vault-240207.internal:8020
jobTracker=instance-1.us-east1-b.c.nifty-vault-240207.internal:8032

queueName=default
oozieExampleRoot=OozieExample

oozie.use.system.libpath=true
oozie.libpath=/user/oozie/share/lib/lib_20190510095907
oozie.wf.application.path=${nameNode}/user/${user.name}/${oozieExampleRoot}/OozieExample_WF

inputPath=${nameNode}/user/${user.name}/data/flume/flume_agent1_data/*.DONE
```

workflow.xml

```
<workflow-app xmlns="uri:oozie:workflow:0.4" name="OozieExample_WF">
   <start to="hive-create-table-node"/>
```

```
   <action name="hive-create-table-node">
      <hive xmlns="uri:oozie:hive-action:0.2">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <job-xml>hive-site.xml</job-xml>
         <configuration>
            <property>
               <name>mapred.job.queue.name</name>
               <value>${queueName}</value>
            </property>
         </configuration>
         <script>customers_create_table_script.hql</script>
      </hive>
      <ok to="hive-insert-table-node"/>
      <error to="fail"/>
   </action>
   <action name="hive-insert-table-node">
      <hive xmlns="uri:oozie:hive-action:0.2">
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <job-xml>hive-site.xml</job-xml>
         <configuration>
            <property>
               <name>mapred.job.queue.name</name>
               <value>${queueName}</value>
            </property>
         </configuration>
         <script>customers_insert_table_script.hql</script>
         <param>INPUT_PATH=${inputPath}</param>
      </hive>
      <ok to="end"/>
      <error to="fail"/>
   </action>
   <kill name="fail">
      <message>Oozie workflow for Hive failed, error
message[${wf:errorMessage(wf:lastErrorNode())}]</message>
   </kill>
   <end name="end"/>
</workflow-app>
```

Command to execute Oozie workflow:

```
oozie job -oozie http://instance-1.us-east1-b.c.nifty-vault-240207.internal:11000/oozie -config
/home/hadoop/oozie_workarea/OozieExample_WF/job.properties -run
```

customers_data_1.txt

```
1,Ramesh,32,Ahmedabad,2000.00
2,Khilan,25,Delhi,1500.00
3,Kaushik,23,Kota,2000.00
4,Chaitali,25,Mumbai,6500.00
```

```
5,Hardik,27,Bhopal,8500.00
6,Komal,22,MP,4500.00
7,Muffy,24,Indore,10000.00
```

customers_data_2.txt

```
8,Jai,32,Chennai,7000.00
9,Siva,25,Delhi,15000.00
10,John,23,Goa,20000.00
11,Martin,25,Chennai,6500.00
12,Joesh,27,Goa,8500.00
13,Kamal,22,Hyderabad,9500.00
14,Rajini,24,Bangalore,10000.00
```

Oozie Coordinator Example

coordinator.properties

```
nameNode=hdfs://instance-1.us-east1-b.c.nifty-vault-240207.internal:8020
jobTracker=instance-1.us-east1-b.c.nifty-vault-240207.internal:8032

queueName=default
oozieExampleRoot=OozieExample

oozieCoordExampleRoot=OozieCoordExample

oozie.use.system.libpath=true
oozie.libpath=/user/oozie/share/lib/lib_20190510095907

oozie.coord.application.path=${nameNode}/user/${user.name}/${oozieCoordExampleRoot}/OozieExample_Coord

workflowPath=${nameNode}/user/${user.name}/${oozieExampleRoot}/OozieExample_WF
inputPath=${nameNode}/user/${user.name}/data/flume/flume_agent1_data/*.DONE

startTime=2019-05-11T02:48Z
endTime=2019-12-31T23:59Z
frequency_min=5
timezone=UTC
```

coordinator.xml

```
<coordinator-app end="${endTime}" frequency="${coord:minutes(frequency_min)}"
name="OozieExample_Coord" start="${startTime}" timezone="${timezone}"
xmlns="uri:oozie:coordinator:0.1">
  <action>
    <workflow>
      <app-path>${workflowPath}</app-path>
    </workflow>
  </action>
```

```
</coordinator-app>
```

Command to execute Oozie coordinator:

```
oozie job -oozie http://instance-1.us-east1-b.c.nifty-vault-240207.internal:11000/oozie -config
/home/hadoop/oozie_workarea/OozieExample_Coord/coordinator.properties -run
```

Reference:

https://oozie.apache.org/docs/5.1.0/index.html#Developer_Documentation